



## Implementando el controlador

### Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video sobre servlet. En el video pasado les dije que quería crear un único servlet que sirva como entrada para poder desde ahí redireccionar a lo que realmente tenemos que hacer y entonces mi idea sería algo como esto, parecido con esto. Ya lo vamos a modificar un poco después, pero bueno, la idea es tener un único servlet.

[00:24] Voy a poner un puntero láser, un único servlet, y que nosotros hagamos todas las requests a ese servlet. Entonces vamos a comenzar haciendo eso, vamos acá a nuestro paquete de servlet y vamos a hacer clic derecho new servlet, vamos a colocarle única entrada servlet, ya vamos a mejorar el nombre después, next.

[00:56] Vamos a decirle que nuestra URL no quiero que sea eso, sino que sea algo más simple entrada. Esa va a ser nuestra URL. Next. Y no quiero hacer doGet y doPost, quiero atender service. Va a atender tanto get como post en un único método y no queremos nuestro constructor. Finish. Vemos que acá nos crea nuestra clase, bien. Una cosa que me gustaría hacer, es agregar nuestro webServlet con nuestra entrada ("/entrada").

[01:37] Y quiero sacarlo de nuestro web.xml. Entonces voy a buscar acá "Ctrl + F" entrada. Bien, encontramos acá, esto de acá no lo quiero, ya después vamos a ver por qué, pero vamos a simplificarlo y dejarlo así. Bien. Eliminé todo lo que tiene que ver con la entrada. Bien, voy a guardar acá y ¿qué es lo que nosotros vamos a tener que hacer?

[02:02] Volviendo a la presentación, yo en realidad lo que quiero es que yo creo un único servlet y ahora tengo que hacer que toda petición que vamos a hacer sea hacia ese servlet. ¿Cómo podemos hacer? Este `listaEmpresas`, `mostrarEmpresa`, `eliminarEmpresa`, en realidad están yendo para servlets diferentes, son URLs diferentes, entonces van a ser diferentes.

[02:25] En este caso, yo quiero convertirlo a algo así. Convertir a `/entrada`, pero el problema es que cada una de esas entradas tiene que poder diferenciarse entre ellas, o sea, tiene que haber una entrada que sirva para listar empresas, una entrada que sirva para mostrar empresas y una entrada que sirva para eliminar empresas.

[02:45] Entonces, ¿cómo es que yo tengo planeado hacer eso? Con lo siguiente. Vamos a usar un parámetro, o sea, va a ser `/entrada`, pero vamos a adicionar un parámetro llamado acción, y ahí nosotros vamos a colocar qué es lo que queremos hacer: listar empresas, mostrar empresas o eliminar empresas, bien. Entonces vamos a hacer esa modificación dentro de nuestro servlet.

[03:11] ¿Cómo vamos a hacer eso? Primero tenemos que leer la entrada. ¿Cómo hacemos para leer una entrada? Vamos a obtener un string. Va a ser `param = request.getParameter`. ¿Y cómo se llama nuestra llave de este servlet? En este caso sería (“acción”). Vamos a leer eso bien, perfecto ahora. ¿Qué es lo que vamos a hacer con ese parámetro?

[03:53] Bueno, nuestra primera idea así, sacándolo rápido de nuestra cabeza sería hacer un `if`. Vamos a comenzar haciendo eso: `if`.

`(param.equals("ListaEmpresas"))` por ejemplo, si es eso, entonces vamos a hacer en este caso un `system.out` de “ListandoEmpresas”. ¿Qué ocurre si no es `listaEmpresas`? Else y vamos a copiar esto de acá solo para hacer más rápido nuestro trabajo.

[04:37] If si es igual a (“MostrarEmpresa”) en singular `MostrarEmpresas`, vamos a decir, (“Mostrando una empresa”) Y ahora un `else if` de si es igual a

("EliminarEmpresa"), que es igual a ("EliminarEmpresa") entonces ("Eliminando una empresa") Eso sería nuestra primera idea así, rápida de que cómo podemos hacer para diferenciar esto.

[05:11] Vamos a probarlo, ya que estamos. Vamos a dar un run y vamos a, bien, no hubo ningún problema. Perfecto, vamos a probar, bueno, ya está, ya lo tengo acá. Enter. Listo las empresas, vamos a ver si en el server aparece alguna cosa dentro de la consola. No está apareciendo nada. Dejen ver que a ver si recargó, sí. Sí recargó. Run. Parece que está funcionando, entonces vamos a ver primero gerenciador.

[05:51] Ah, claro, el problema es que es entrada: entrada? acción=listaEmpresas. Vale, era esto. No muestra nada, pero no dio error y está listando empresas. Entonces está llegando acá, bien. Esto se está mostrando, perfecto. Lo mismo va a ocurrir con los otros, bien, perfecto. Ya tenemos una cosa resuelta, ¿ahora qué es lo que ocurre? Nosotros tenemos que hacer que funcione.

[06:24] Si estamos entrando en listarEmpresa, nosotros queremos, obviamente listar empresas, entonces también nuestra primera idea sería vamos a la listaEmpresasServlet, copiamos esto de acá "Ctrl + C" y lo metemos acá en dentro del if, "Ctrl + V". Eso a priori parece que está funcionando, vamos a dar un run de nuevo. ListarEmpresas, entrar y parece que está funcionando. Bien.

[06:55] ¿Pero qué es lo que ocurre? Esa metodología, esa idea de copiar y pegar no es muy bueno. Imagínense, cuando tengamos acá todos nuestros servlets, todo lo que tienen que hacer nuestras diferentes clases, estaría llenísimo de código acá. Eso no queda bien, no es bueno y no es muy fácil de mantener. ¿Entonces, qué es lo que vamos a hacer?

[07:25] Lo que nosotros vamos a hacer es crear una clase que sirva específicamente para hacer eso, listar empresas. ¿Entonces cómo hacemos eso? Vamos a ir a nuestro src/main/java, clic derecho, new class, vamos a

llamar acá de `listaEmpresas`, no va a ser un `servlet`, esto va a ser una clase que hace eso y el paquete que vamos a colocar va a ser uno nuevo.

[07:54] Va a ser `com.alura.gerenciador.accion`. Bien, esa va a ser la idea. Finish y creamos una clase. Perfecto. ¿Ahora, qué es lo que nosotros tenemos que hacer? Yo lo que quiero hacer ahora es que desde nuestra única entrada `servlet` llamar a esa clase y que esa clase sea la que ejecute todo esto de listar empresas.

[08:26] Entonces lo que voy a hacer es dentro de `ListaEmpresas` voy a crear una función `public void`. Voy a llamarla de ejecutar. Y voy a copiar el código que tenemos acá adentro allá, vamos a colocar incluido el `system.out`, "`Ctrl + X`" para cortarlo, "`Ctrl + V`". Bien copiamos el código, esta clase debería poder hacer lo mismo. Perdón. Este método en realidad debería poder hacer lo mismo que estaba haciendo nuestro `servlet` de listar empresas. ¿Qué es lo que ocurre?

[09:08] Acá tenemos algún problema que no tenemos nuestro `request` y no tenemos nuestro `response`. ¿Entonces cómo hacemos para que ese método tenga esos objetos de `request` y `response`? Vamos a enviárselos. Desde nuestro `if` vamos a hacer, `if param`, bla bla bla, vamos a crear nuestra lista de empresas: `ListaEmpresas`, vamos a crear una instancia. Acción. ¿Cómo podría poner?

[09:41] Sí. Le voy a llamar de acción, igual a `new ListaEmpresas()`; y acá haría un acción. Voy a importar, "`Ctrl + Shift + O`", importé nuestra `ListaEmpresas`, `acción.ejecutar`, y acá yo quiero mandarle los parámetros, entonces le mandaría nuestro `request` y nuestro `response`. Perfecto. "`Ctrl + Shift + O`".

[10:12] Nos está diciendo ejecutar no tiene esos parámetros. Okay, vamos a ir a nuestra lista de empresas y vamos a cambiar acá. Los parámetros van a ser `HttpServletRequest`. Bien. Vamos a llamarlo de `request` y `HttpServletResponse`. Acá nosotros tenemos que, perfecto nos está diciendo que tenemos que poner una `throw exception`.

[10:51] En caso de que dé algún error tenemos que colocarlo. Entonces, colocando el mouse arriba de esa parte señalada en rojo, add throw declaration, enter, enter. Perfecto. Nosotros tenemos ya nuestro método ejecutar que en teoría hace lo mismo que estaba haciendo nuestro servlet de listar empresas.

[11:13] "Ctrl + S", no se olviden, vamos a reiniciar el servidor porque ya hice varios cambios bien y ahora vamos a probar, vamos a ver si eso funciona. Tenemos una acción listaEmpresa, enter. Nos está diciendo listado empresas y funcionó. Sí, funcionó. Perfecto.

[11:36] Entonces, ahora en este momento ya no estoy colocando toda mi lógica de servlet ahí dentro de ese if, dentro de este única entrada servlet y sí lo estoy dividiendo y lo estoy llevando a una clase. Atención, esto no es un servlet. Este listaEmpresas no es un servlet, fíjense que no tiene una herencia de, por ejemplo, httpServlet no tiene un extend de servlet. Esto no es un servlet, es algo bien más liviano, bien más tranquilo.

[12:08] Bien, ahora que está funcionando quiero dar una reforma acá en el código. Acá voy a llamarle paramAcción, porque estamos obteniendo el parámetro acción específicamente. "Ctrl + C" "Ctrl + V" acá en param, "Ctrl + S".

[12:29] Con esto nosotros ya tenemos funcionando entonces nuestro pequeño código, nuestro único servlet junto con nuestra clase que va a ejecutar el código que antes hacía nuestro listadorEmpresasServlet. Bien. ¿Qué es lo que faltaría? Faltaría hacer la misma cosa pero con mostrarEmpresa y eliminarEmpresas.

[12:52] Entonces les dejo esa pequeña tarea a ustedes, si no consiguen hacerlo, bueno, los espero en el próximo video que ya vamos a hacer eso. Nos vemos allá.

